

AMENDMENTS TO THE CLAIMS

Claims 1-34 are pending in the instant application. Claims 2-5, 7-10, 12-15, and 17-20 have been amended for clarification. Claims 6, 16 and 26 are cancelled and have been replaced by added claims 32-34 respectively. The Applicant requests reconsideration of the claims in view of the following amendments reflected in the listing of claims.

Listing of claims:

1. (Original) A method for posting buffers for a non-posting TCP application, the method comprising:

posting at least one generic buffer located in a memory external to a host adapter; and

transferring incoming data for a TCP connection to said at least one posted generic buffer prior to the non-posting TCP application posting a TCP application posted buffer for said incoming data.

2. (Amended) The method according to claim 1, ~~further~~ comprising allocating at least one generic buffer from a pool of available generic buffers upon receipt of said incoming data for said TCP connection.

3. (Amended) The method according to claim 2, ~~further~~ comprising storing at least a portion of said incoming data in said allocated at least one generic buffer if said TCP application posted buffer is unable to accommodate said incoming data.

4. (Amended) The method according to claim 1, ~~further~~ comprising determining whether said incoming data for said TCP connection transferred to said at least one posted generic buffer is in sequence.

5. (Amended) The method according to claim 4, wherein said determining ~~further~~ comprises assembling said incoming data in said at least one posted generic buffer if said incoming data is out of sequence.

6. (Cancelled)

7. (Amended) The method according to claim 1, ~~further~~ comprising polling said at least one posted generic buffer by the non-posting TCP application to determine when said at least one posted generic buffer contains data for said TCP connection.

8. (Amended) The method according to claim 7, ~~further~~ comprising, in response to said polling, notifying the non-posting TCP application when said incoming data is stored in said at least one posted generic buffer.

9. (Amended) The method according to claim 1, ~~further~~ comprising posting at least one generic buffer prior to said transferring of said incoming data for a TCP connection to at least a portion of available TCP application buffers.

10. (Amended) The method according to claim 1, ~~further~~ comprising posting at least one posted generic buffer located in a memory internal to said host adapter.

11. (Original) A machine-readable storage, having stored thereon a computer program having at least one code section for posting buffers for a non-posting TCP application, the at least one code section executable by a machine for causing the machine to perform the steps comprising:

posting at least one generic buffer located in a memory external to a host adapter; and

transferring incoming data for a TCP connection to said at least one posted generic buffer prior to the non-posting TCP application posting a TCP application posted buffer for said incoming data.

12. (Amended) The machine-readable storage according to claim 11, ~~further~~ comprising code for allocating at least one generic buffer from a pool of available generic buffers upon receipt of said incoming data for said TCP connection.

13. (Amended) The machine-readable storage according to claim 12, ~~further~~ comprising code for storing at least a portion of said incoming data in said allocated at least one generic buffer if said TCP application posted buffer is unable to accommodate said incoming data.

14. (Amended) The machine-readable storage according to claim 11, ~~further~~ comprising code for determining whether said incoming data for said TCP connection transferred to said at least one posted generic buffer is in sequence.

15. (Amended) The machine-readable storage according to claim 14, ~~further~~ comprising code for assembling said incoming data in said at least one posted generic buffer if said incoming data is out of sequence.

16. (Cancelled)

17. (Amended) The machine-readable storage according to claim 11, ~~further~~ comprising code for polling said at least one posted generic buffer by the non-posting TCP application to determine when said at least one posted generic buffer contains data for said TCP connection.

18. (Amended) The machine-readable storage according to claim 17, ~~further~~ comprising code for notifying the non-posting TCP application when said incoming data is stored in said at least one posted generic buffer in response to said polling.

19. (Amended) The machine-readable storage according to claim 11, ~~further~~ comprising code for posting at least one generic buffer prior to said transferring of said incoming data for a TCP connection to at least a portion of available TCP application buffers.

20. (Amended) The machine-readable storage according to claim 11, ~~further~~ comprising code for posting at least one posted generic buffer located in a memory internal to said host adapter.

21. (Original) A system for posting buffers for a non-posting TCP application, the system comprising:

at least one generic buffer located in a memory external to a host adapter;

at least one processor that posts said at least one generic buffer located in said memory external to said host adapter; and

said at least one processor transfers incoming data for a TCP connection to said at least one posted generic buffer prior to the non-posting TCP application posting a TCP application posted buffer for said incoming data.

22. (Original) The system according to claim 21, wherein said at least one processor allocates at least one generic buffer from a pool of available generic buffers upon receipt of said incoming data for said TCP connection.

23. (Original) The system according to claim 22, wherein said at least one processor stores at least a portion of said incoming data in said allocated at least one generic buffer if said TCP application posted buffer is unable to accommodate said incoming data.

24. (Original) The system according to claim 21, wherein said at least one processor determines whether said incoming data for said TCP connection transferred to said at least one posted generic buffer is in sequence.

25. (Original) The system according to claim 24, wherein said at least one processor assembles said incoming data in said at least one posted generic buffer if said incoming data is out of sequence.

26. (Cancelled)

27. (Original) The system according to claim 21, wherein said at least one processor polls said at least one posted generic buffer by the non-posting TCP application to determine when said at least one posted generic buffer contains data for said TCP connection.

28. (Original) The system according to claim 27, wherein said at least one processor notifies the non-posting TCP application when said incoming data is stored in said at least one posted generic buffer in response to said polling.

29. (Original) The system according to claim 21, wherein said at least one processor posts at least one generic buffer prior to said transferring of said incoming data for a TCP connection to at least a portion of available TCP application buffers.

30. (Original) The system according to claim 21, wherein said at least one processor posts at least one posted generic buffer located in a memory internal to said host adapter.

31. (Original) The system according to claim 21, wherein said at least one processor is at least one of a NIC, a generic buffer handler, a buffer post handler, an application, a TCP notifier and a TCP data placement processor.

32. (New) A method for posting buffers for a non-posting TCP application, the method comprising:

posting at least one generic buffer located in a memory external to a host adapter;

transferring incoming data for a TCP connection to said at least one posted generic buffer prior to the non-posting TCP application posting a TCP application posted buffer for said incoming data;

determining whether said incoming data for said TCP connection transferred to said at least one posted generic buffer is in sequence;

assembling said incoming data in said at least one posted generic buffer if said incoming data is out of sequence; and

ordering said at least one posted generic buffer using a TCP data placement sequence number for said out of sequence incoming data.

33. (New) A machine-readable storage, having stored thereon a computer program having at least one code section for posting buffers for a non-posting TCP application, the at least one code section executable by a machine for causing the machine to perform the steps comprising:

posting at least one generic buffer located in a memory external to a host adapter;

transferring incoming data for a TCP connection to said at least one posted generic buffer prior to the non-posting TCP application posting a TCP application posted buffer for said incoming data;

determining whether said incoming data for said TCP connection transferred to said at least one posted generic buffer is in sequence;

assembling said incoming data in said at least one posted generic buffer if said incoming data is out of sequence; and

ordering said at least one posted generic buffer using a TCP data placement sequence number for said out of sequence incoming data.

34. (New) A system for posting buffers for a non-posting TCP application, the system comprising:

at least one generic buffer located in a memory external to a host adapter;

at least one processor that posts said at least one generic buffer located in said memory external to said host adapter;

said at least one processor transfers incoming data for a TCP connection to said at least one posted generic buffer prior to the non-posting TCP application posting a TCP application posted buffer for said incoming data; and

wherein said at least one processor determines whether said incoming data for said TCP connection transferred to said at least one posted generic buffer is in sequence, wherein:

said at least one processor assembles said incoming data in said at least one posted generic buffer if said incoming data is out of sequence; and

said at least one processor orders said at least one posted generic buffer using a TCP data placement sequence number for said out of sequence incoming data.